What is claimed is:

[Claim 1] A system of storing data, in computer memory, associated to the description of any arbitrarily defined data structure, with nested data structures to any level, with optional plurality.

[Claim 2] The system of claim 1, further comprising: a method for setting the data values of any data element within the structure.

[Claim 3] The system of claim 1, further comprising: a method for enforcing the integrity of data while and after data is set.

[Claim 4] The method of claim 3, further comprising: a method of enforcing software interface integrity, data type validation and enforcing data type restrictions at runtime.

[Claim 5] The system of claim 1, further comprising: a method for getting the data values of any data element within the structure.

[Claim 6] A method of describing data mapping/connectivity between two separate and optionally nested and arbitrarily defined data structures with optional plurality.

[Claim 7] A method of claim 6, further comprising: a method of automated transfer of data between two, or more data structures.

[Claim 8] A method of claim 6, further comprising: describing data mapping between the outputs or inputs of one or more software services to the inputs or outputs of one or more software services, including but not limited to Web services.

[Claim 9] A method of claim 7, further comprising: automated transfer of data between two, or more connected software services including but not limited to Web services.

[Claim 10] A method of claim 6, further comprising: describing data mapping between input/outputs of one or more software services and a programming construct, such as but not limited to a data-driven, semantic-based branching construct.

[Claim 11] A method of claim 7, further comprising: automated transfer of data between one or more software services and a programming construct.

[Claim 12] All other methods and systems explicit or implicit in the description of the enclosed invention.